



Oracle9i™ for Windows® 2000: Backup and Recovery Using a NetApp® Controller in a SAN Environment

Toby Creek, NetApp | TR-3211

ARCHIVAL COPY
Contents may be out-of-date

TECHNICAL
REPORT

NetApp, a pioneer and industry leader in data storage technology, helps organizations understand and meet complex technical challenges with advanced storage solutions and global data management strategies.

Table of Contents

1. [Purpose and Scope](#)
 2. [Key Advantages](#)
 3. [Requirements and Assumptions](#)
 4. [Backup](#)
 - 4.1. [General Backup Concerns with Fibre Channel](#)
 - 4.2. [Cold Backup](#)
 - 4.3. [Hot Backup](#)
 - 4.4. [Backup using Writable Snapshots](#)
 5. [Restore](#)
 - 5.1. [Restoring a Single Data File within a File System From a Writable LUN](#)
 - 5.2. [Restoring All Data Files in a Single File System or a Raw Device](#)
 - 5.3. [Restoring a Database That Was Run in Noarchivelog Mode](#)
 - 5.4. [Restoring a Database That Was Run in Archivelog Mode](#)
 6. [Conclusions](#)
 7. [Caveats](#)
-

1. Purpose and Scope

This document covers the techniques for backing up and restoring an Oracle9i for Windows 2000 database when a NetApp™ Fibre Channel-attached controller is used for database storage. Specifically, this report covers the following issues:

- Backing up a database while the database is shut down ("cold" backup)
- Backing up a database while the database is running ("hot" backup)
- Recovering a single file from a file system on a LUN
- Recovering a single file system or raw device
- Recovering a database that failed while running in noarchivelog mode
- Recovering a database that failed while running archivelog mode
- Recovering a database after the loss of the hardware environment (DR)

2. Key Advantages

The performance and reliability of the backup and recovery operations are critical to effective database operation. NetApp provides unique functionality that affords the database administrator significant reliability and performance benefits in both backup and recovery. The key feature of NetApp's approach to database backup is the use of Snapshot™ technology. For more information

on how Snapshots work, see File System Design for an NFS File Server Appliance by Dave Hitz, James Lau, and Michael Malcolm (http://www.netapp.com/tech_library/3002.html). Those with NOW™ ([NetApp on the Web](#)) access can also check the following NetApp documentation, which covers Snapshots in detail. (NetApp on the Web is NetApp's online support site. This site is open to all current NetApp customers.)

Snapshots are critical because they allow the database administrator to quickly and easily create an image of all LUNs and associated file systems, including the database files. Once the Snapshot is taken, the database can be returned to normal operation. A backup can then be written to tape from the Snapshot directory while the database is being used from the active file system. Actual backup-to-tape performance is thus of secondary importance, since the impact of this operation on the database's performance is negligible. For the backup operation, the paramount issue is that the database must either be shutdown and restarted in the minimal amount of time ("cold" backup) or remain in "hot-backup" mode for the minimal amount of time ("hot" backup).

During a cold backup, the database is shut down and unavailable. Obviously, reducing the period of time during which this is the case provides an advantage.

During a hot backup, the database is online, but in hot-backup mode. This results in significant performance overhead, particularly in write operations. Further, being in hot backup mode can result in difficulties should the database crash. Thus, getting the database out of hot-backup mode more quickly also provides an advantage.

Since the time required for the backup operation is reduced so much, many database administrators may find offline or cold backups become feasible when running a database with storage on a controller. Many databases cannot be shut down for the hours required to perform an offline backup; however, shutting down the database for only a few seconds may be acceptable. The functionality of Snapshots affords the database administrator the ability to shut down the database for only a few seconds, during which a Snapshot is taken, and then bring the database back up again. Cold backups are preferable to hot backups for many reasons.

Using hot backups, you can also take Snapshots several times per day and back up only one of them to tape. This offers the database administrator additional flexibility. Effectively, you can take an online backup of your entire database every few hours. (Note: The amount of storage overhead associated with a Snapshot is based upon the number of blocks that are different between the Snapshot and the active file system. As this increases, the size of the Snapshot's storage requirements will increase as well. Thus, recent Snapshots are generally less onerous to store than older ones.)

Snapshots also provide key advantages for the restore operation. If the data that you need is in a Snapshot, you can restore the entire database, regardless of size, in a few minutes. This is accomplished with the Data ONTAP™ "snap restore" command. If you save several days' worth of Snapshots, the chances are that you will never need to restore from tape at all, barring a failure on the controller itself. (Note: NetApp always recommends that all data be backed up to tape often. Snapshots should be used as a supplement to, not a replacement for, tape backup.)

Take the following scenario. A company is running a 200GB database on a Windows 2000 platform, with storage on local disk. A glitch in one of the database server's CPUs results in corrupt data being

written to the data files. As a result, the entire database must be restored from tape. At a rate of about 50GB per hour, it will take approximately four hours to restore the data.

Now consider the same scenario with storage on a controller. Since the database administrator saved a Snapshot from a period prior to the failure, all that is required is to restore the data files from the Snapshot. Using the SnapRestore® ("snap restore") command, that can be accomplished in only a few minutes. This provides vastly improved mean time to recovery over that provided by the local disk solution.

3. Requirements and Assumptions

This technical report covers backup and recovery of an Oracle9i database running under Windows 2000 with database storage on a NetApp Fibre Channel-connected controller. It is assumed that the reader is familiar with Oracle9i and the operation of NetApp controllers. It is also assumed that the reader is familiar with the operation and administration of Microsoft® Windows 2000. All examples in this technical document are tested on Oracle9i Enterprise Edition under Microsoft Windows 2000 with Service Pack 2 applied. The scripts contained in this paper may require significant modifications to run in later versions of Windows. Also, the scripts presented are not intended to be "drop-in" backup scripts. They are the minimum required to display the capabilities of the controller, but may omit steps that are considered best practices with Oracle®.

The scripts in this report rely on the existence of the SnapDrive™ command line interface utility. For reasons explained later in this document, the SnapDrive interface must be used to obtain consistent backups. The examples in this technical report use the following syntax:

```
sdcli snap Command -s SnapshotName -d DriveLetter
```

For Snapshot management, the SnapDrive command line interface provides the functionality to rename Snapshots. The syntax in this instance would look like the following:

```
sdcli snap rename -d DriveLetter -o OldName -n NewName
```

The sample scripts in this technical report assume the following:

- The name of the controller is "data"
- The location of the database LUNs is "data:/vol/dbvol", as drive X
- The location of the archivelog LUNs is "data:/vol/archvol", as drive Y
- The name of the Oracle instance is "orcl"
- The scripts are stored in the directory "o:\orascripts"
- The SnapDrive command line utilities are installed in "c:\Program Files\SnapDrive"

4. Backup

The following sections contain SQL and operating system scripts that show the techniques for backing up an Oracle database with data files stored on a NetApp controller.

4.1. General Backup Concerns with Fibre Channel

There are several generalities that need to be applied to all backup and recovery solutions using the NetApp controller Snapshots. With the controller acting as a block device, the host carries much more intelligence about the state of the file system than it does in NAS configurations. As a result, the host has to be more integrated with the Snapshot facilities in the controller to make sure that metadata from the file system is flushed to disk before the Snapshot is taken. Specifically, Snapshots of NTFS file systems should never be performed from the controller command line or via rsh. The sdcli.exe utility included with SnapDrive performs the necessary synchronization for NTFS to get a consistent Snapshot. Note that application-level synchronizations such as Oracle's hot-backup mode are not addressed by SnapDrive, so these still need to be integrated into the backup mechanism.

The reader is strongly encouraged to become familiar with the sdcli utility and all of its options. The proper use of the utility will be critical to ensure consistent backups and successful restores. For more information, see the SnapDrive 2.0 documentation.

4.2. Cold Backup

If the database is run in noarchivelog mode, the only option is to take an offline or cold backup. A cold backup is often periodically taken for a database that is run in archivelog mode as well. A cold backup requires that the database administrator shut down the database, back up all critical files using operating-system commands, and then restart the database.

Taking a cold backup using a NetApp controller is a simple and fast operation. It requires three scripts, which have the following functions:

Script Name	Function
shutdown.sql	Shuts down the Oracle instance.
startup.sql	Starts up the Oracle instance.
docoldbackup.bat	A wrapper batch script that calls shutdown.sql, takes a Snapshot, calls startup.sql, and performs the backup -to tape from the Snapshot. This script also does some housecleaning to keep four Snapshots online at a time. If the script is run every night as a scheduled task, that means that four days' worth of Snapshots would be available at any given time. For more information, see 2. Key Advantages.

The following is the text of the shutdown.sql script:

```
CONNECT sys/change_on_install as sysdba
SHUTDOWN IMMEDIATE
EXIT
```

You may want to wrap your "SHUTDOWN IMMEDIATE" command inside a batch file that provides some warning and error processing. For more information on how to do this, see the documentation that came with your version of Oracle. The following is the text of the startup.sql script:

```
CONNECT sys/change_on_install as sysdba
STARTUP
EXIT
```

The following is the text of the docoldbackup.bat shell script:

```
@echo off
path=%PATH%;C:\Program Files\SnapDrive

: Rename and delete old Snapshots
sdcli snap delete -d X -s old3
sdcli snap rename -d X -o old2 -n old3
sdcli snap rename -d X -o old1 -n old2
sdcli snap rename -d X -o new -n old1

: Shutdown the Oracle instance
sqlplus /nolog @o:\orascripts\shutdown.sql

: Take a new Snapshot
sdcli snap create -s new -D X

: Startup the Oracle instance again
sqlplus /nolog @o:\orascripts\startup.sql

: Perform your backup using dump
: or other operating system commands here.
: For example:
: rsh -l root data dump 0u1bn rst0a 63
/vol/voldb/home/oracle/.snapshot/new/
```

The options for dumping data from a controller to tape are beyond the scope of this technical report. However, they are amply documented in the *NetApp Controller Systems Administrator's Guide*.

Executing the docoldbackup batch script will look something like the following:

```
O:> o:\orascripts\docoldbackup.bat

SQL*Plus: Release 9.0.1.0.1 - Production on Wed Jul 24 14:11:55 2002

(c) Copyright 2001 Oracle Corporation. All rights reserved.

Connected.
Database closed.
Database dismounted.
ORACLE instance shut down.
Disconnected from Oracle9i Enterprise Edition Release 9.0.1.1.1 - Production
With the Partitioning option
JServer Release 9.0.1.1.1 - Production
Initializing...
KIM : Checking Policies ...
KIM : Ready to create Snapshot

SQL*Plus: Release 9.0.1.0.1 - Production on Wed Jul 24 14:12:11 2002
```

(c) Copyright 2001 Oracle Corporation. All rights reserved.

Connected to an idle instance.
ORACLE instance started.

```
Total System Global Area 218918824 bytes
Fixed Size                 282536 bytes
Variable Size              159383552 bytes
Database Buffers          58720256 bytes
Redo Buffers               532480 bytes
Database mounted.
Database opened.
Disconnected from Oracle9i Enterprise Edition Release 9.0.1.1.1 - Production
With the Partitioning option
JServer Release 9.0.1.1.1 - Production
```

This process should take about one minute. *Note:* The size of the database will not appreciably affect this interval, since the time required to take a Snapshot is not dependent upon the size of the file system or the files within the file system. Rather, a Snapshot operation triggers a consistency point within the controller and then simply copies the master inode of the file system to a new location. Thus, only about 4Kb of data is actually copied. For more information see File System Design for an NFS File Server Appliance by Dave Hitz, James Lau, and Michael Malcolm (http://www.netapp.com/tech_library/3002.html).

4.3. Hot Backup

If the database is run in archivelog mode, you have the option to take online or hot backups. The database is open for user access while a hot backup is running. However, there is a significant performance hit, and an increased level of redo activity is generated. For this reason, it is critical that the database remain in hot-backup mode for the absolute minimum period of time possible. NetApp affords the database administrator the opportunity to reduce the interval during which the database is in hot-backup mode to only a few minutes.

Since a database in hot-backup mode can be restored up to the point of failure in most instances, you should plan for recovery by organizing your files differently on the controller. The configuration recommended by NetApp for Oracle databases running in archivelog mode is to place the data files on one volume and the control file and log files (including the archived log files) on a second volume. This enables you to recover the data files using the "snap restore" command while maintaining the control and log files intact. Once you apply the log files, your database is current up to the moment of failure. For more information on this, see [5.4. Restoring a Database That Was Run in Archivelog Mode](#).

There is less margin for error when taking a hot backup. Since a cold backup using a NetApp controller takes only about one minute, a cold backup may be a feasible solution for many databases. For more information see [4.2. Cold Backup](#).

Taking a hot backup using a NetApp controller is a simple and fast operation. It requires five scripts, two of which are created during the operation. These scripts have the following functions:

Script Name	Function
-------------	----------

dobegin.sql	Writes out the begin.sql script, and then calls it.
begin.sql	Places all of the tablespaces that need to be backed up into hot-backup mode.
doend.sql	Writes out the end.sql script, and then calls it.
end.sql	Takes all of the tablespaces that need to be backed up out of hot-backup mode.
dohotbackup.bat	A wrapper batch script that calls dobegin.sql, takes a Snapshot, calls doend.sql, and performs the backup to tape from the Snapshot. This script also does some housecleaning to keep four Snapshots online. If the script is run every night as a scheduled task, that means that four days' worth of Snapshots would be available at any given time. Alternatively, you may want to take a hot backup more frequently than once a day. For more information, see 2. Key Advantages .

The following is the text of the dobegin.sql script:

```
CONNECT system/manager
SET FEEDBACK off
SET PAGESIZE 0
SPOOL o:\orascripts\begin.sql
SELECT
  'ALTER TABLESPACE ' ||
  tablespace_name ||
  ' BEGIN BACKUP;'
FROM
  dba_tablespaces;
SPOOL off
@begin
EXIT
```

On our system (admittedly a small test database), the following begin.sql script is generated by dobegin.sql:

```
ALTER TABLESPACE SYSTEM BEGIN BACKUP;
ALTER TABLESPACE UNDOTBS BEGIN BACKUP;
ALTER TABLESPACE CWMRITE BEGIN BACKUP;
ALTER TABLESPACE DRSYS BEGIN BACKUP;
ALTER TABLESPACE INDX BEGIN BACKUP;
ALTER TABLESPACE TEMP BEGIN BACKUP;
ALTER TABLESPACE TOOLS BEGIN BACKUP;
ALTER TABLESPACE USERS BEGIN BACKUP;
ALTER TABLESPACE TEMP BEGIN BACKUP;
```

The following is the text of the doend.sql script:

```
CONNECT system/manager
SET FEEDBACK off
SET PAGESIZE 0
SPOOL o:\orascripts\end.sql
SELECT
  'ALTER TABLESPACE ' ||
  tablespace_name ||
  ' END BACKUP;'
FROM
  dba_tablespaces;
SPOOL off
@end
EXIT
```


On our system the following end.sql script is generated by doend.sql:

```
ALTER TABLESPACE SYSTEM END BACKUP;
ALTER TABLESPACE UNDOTBS END BACKUP;
ALTER TABLESPACE CWMLITE END BACKUP;
ALTER TABLESPACE DRSYS END BACKUP;
ALTER TABLESPACE INDX END BACKUP;
ALTER TABLESPACE TEMP END BACKUP;
ALTER TABLESPACE TOOLS END BACKUP;
ALTER TABLESPACE USERS END BACKUP;
ALTER TABLESPACE TEMP END BACKUP;
```

The following is text of the dohotbackup.bat script:

```
@echo off
set PATH=%PATH%;C:\Program Files\SnapDrive
set ORACLE_SID=orcl

: Rename and delete old Snapshots
sdcli snap delete -d X -s old3
sdcli snap rename -d X -o old2 -n old3
sdcli snap rename -d X -o old1 -n old2
sdcli snap rename -d X -o new -n old1

: Place all of the critical tablespaces in
: hot backup mode.
sqlplus system/manager @o:\orascripts\dobegin.sql

: Take a new Snapshot
sdcli snap create -s new -D X

: Remove all affected tablespaces from hot
: backup mode
sqlplus system/manager @o:\orascripts\doend.sql

: Perform your backup using dump
: or other operating system commands here.
: For example:
: rsh -l root data dump 0ufbln rst0a 63
/vol/dbvol/home/oracle/.snapshot/new/
```

The options for dumping data from a controller to tape are beyond the scope of this technical report. They are amply documented in the *NetApp Controller Systems Administrator's Guide*.

At that point, calling the dohotbackup batch script will do the following:

```
O:> o:\orascripts\dohotbackup.bat

SQL*Plus: Release 9.0.1.0.1 - Production on Wed Jul 24 14:01:10 2002
```

(c) Copyright 2001 Oracle Corporation. All rights reserved.
Connected to:
Oracle9i Enterprise Edition Release 9.0.1.1.1 - Production
With the Partitioning option
JServer Release 9.0.1.1.1 - Production

Connected.

```
ALTER TABLESPACE SYSTEM BEGIN BACKUP;  
ALTER TABLESPACE UNDOTBS BEGIN BACKUP;  
ALTER TABLESPACE CWMLITE BEGIN BACKUP;  
ALTER TABLESPACE DRSYS BEGIN BACKUP;  
ALTER TABLESPACE INDX BEGIN BACKUP;  
ALTER TABLESPACE TEMP BEGIN BACKUP;  
ALTER TABLESPACE TOOLS BEGIN BACKUP;  
ALTER TABLESPACE USERS BEGIN BACKUP;  
ALTER TABLESPACE TEMP BEGIN BACKUP
```

*

ERROR at line 1:

ORA-03217: invalid option for alter of TEMPORARY TABLESPACE

Disconnected from Oracle9i Enterprise Edition Release 9.0.1.1.1 -
Production

With the Partitioning option
JServer Release 9.0.1.1.1 - Production

Initializing...

KIM : Checking Policies ...
KIM : Ready to create Snapshot

SQL*Plus: Release 9.0.1.0.1 - Production on Wed Jul 24 14:01:19 2002

(c) Copyright 2001 Oracle Corporation. All rights reserved.

Connected to:

Oracle9i Enterprise Edition Release 9.0.1.1.1 - Production
With the Partitioning option
JServer Release 9.0.1.1.1 - Production

Connected.

```
ALTER TABLESPACE SYSTEM END BACKUP;  
ALTER TABLESPACE UNDOTBS END BACKUP;  
ALTER TABLESPACE CWMLITE END BACKUP;  
ALTER TABLESPACE DRSYS END BACKUP;  
ALTER TABLESPACE INDX END BACKUP;  
ALTER TABLESPACE TEMP END BACKUP;  
ALTER TABLESPACE TOOLS END BACKUP;  
ALTER TABLESPACE USERS END BACKUP;
```

```
ALTER TABLESPACE TEMP END BACKUP
*
ERROR at line 1:
ORA-03217: invalid option for alter of TEMPORARY TABLESPACE

Disconnected from Oracle9i Enterprise Edition Release 9.0.1.1.1 -
Production
With the Partitioning option
JServer Release 9.0.1.1.1 - Production
```

On our system the operation took approximately one minute. *Note:* The size of the database will not appreciably affect this interval, since the time required to take a Snapshot is not dependent upon the size of the file system or the files within the file system. Rather, a Snapshot operation triggers a consistency point within the controller and then simply copies the master inode of the file system to a new location. Thus, only about 4Kb of data is actually copied. For more information see *File System Design for an NFS File Server Appliance* by Dave Hitz, James Lau, and Michael Malcolm (http://www.netapp.com/tech_library/3002.html).

4.4. Backup using Writeable Snapshots

Before snapshots, a common database backup technique was to use a physical mirror of production data, merged, split, and mounted to another host to perform backup operations. Using LUNs created from writeable snapshots closely emulates this technique, replacing the lengthy merge and split cycle with a snapshot. The snapshot is then mounted to the backup host, thereby offloading the processes associated with performing the backup to a completely separate machine.

This backup technique has multiple benefits. The most significant is that the application server's performance is not affected during the backup cycle. Because the backup server is also connected to the Storage Area Network, the data traffic from the controller to the tape array does not impact the local area network interface or the CPU of the application server. Additionally, the data does not have to pass through the network stack of both machines as IP packets, avoiding additional transfer overhead. Lastly, as with traditional snapshots, the actual window required to move the data to tape no longer affects the application.

System requirements for the backup server are similar to the application server from a connectivity and software standpoint. The SnapDrive software must be installed on the backup server. It serves to trigger the snapshot from the application server using Windows Remote Procedure Calls, which will perform the required filesystem buffer cache flushes. For this reason, the backup and application servers must be members of the same domain, or have an appropriate trust relationship defined.

The backup server must also be connected to the SAN and have visibility to the storage subsystem. Zones should be created within the Fibre Channel switch to partition the servers and storage from any Fibre Channel connected tape devices. The backup server will be a member of both zones, so that it may see the storage and the tape library simultaneously. For each concurrently mounted snapshot, the controller must have sufficient space reserved for any read-write LUNs that are used to perform the backup.

The backup software must have the capability to execute a pre-backup script to invoke Oracle's hot backup mode as well as the sdcli command line interface. The mounted snapshot will be presented to the backup server as a local drive. Once mounted, this local drive on the backup server will be the source for the database backup. For this reason, it is recommended that the database backup be configured as a separate job on the backup server. Also, the backup software should allow the restoration of data to an alternate destination from the original source.

A script must be executed before the backup to tape commences. This script will do all of the necessary steps to create and present the snapshot to the backup server.

A fully commented sample pre-backup script is shown below

```
@echo off
set PATH=%PATH%;C:\Program Files\SnapDrive
set ORACLE_SID=orcl
set DBHOST=dbhost

: Unmount, Rename and delete old Snapshots
sdcli snap delete -d X -s old3
sdcli snap rename -d X -o old2 -n old3
sdcli snap rename -d X -o old1 -n old2
sdcli snap rename -d X -o new -n old1

: Place all of the critical tablespaces in
```

```

: hot backup mode.
sqlplus system/manager@%DBHOST% @o:\orascripts\dobegin.sql

: Take a new Snapshot
sdcli snap create -m %DBHOST% -s new -D X

: Remove all affected tablespaces from hot
: backup mode.
sqlplus system/manager@%DBHOST% @o:\orascripts\doend.sql

: Mount the writable snapshot as drive X:
sdcli snap mount -r %DBHOST% -k X -s new -d X
if not ERRORLEVEL 0 exit 1

: Drive X is now ready for backup

exit 0

```

The script below will unmount the snapshot after the backup is complete. With SnapDrive, the unmount operation also destroys the read/write LUN on the controller.

```

@echo off
set PATH=%PATH%;C:\Program Files\SnapDrive
set ORACLE_SID=orcl
set DBHOST=dbhost

: Unmount the snapshot. If it will not unmount cleanly, use the force
option.
sdcli snap unmount -d X
if not ERRORLEVEL 0 sdcli snap unmount -d X -f

exit 0

```

5. Restore

The examples concerning restore in this technical report assume that the data required is still stored in a Snapshot. If that is not the case (either because the Snapshot has been deleted or because the failure was on the controller itself) then you will need to perform a restore-from-tape operation. The requirements to do this are comparable to a similar operation on a Windows 2000 system. The steps are beyond the scope of this paper, but are amply documented in the *NetApp Controller Systems Administrator's Guide*.

5.1. Restoring a Single Data File within a File System from a Writable LUN

The single data file restore using a block-based device with a file system on it is complicated by the fact that the controller does not understand the format of the file system. In this case, it will be necessary to create a writeable LUN from the Snapshot and connect it to a free drive letter or other directory location on the server. The data file can then be copied from the Snapshot location and recovered. Oracle must be running in archivelog mode at the time of the Snapshot to recover successfully.

The steps to perform this type of recovery are outlined below:

1. Create a writeable LUN from the Snapshot and present it to the host. In this example, we are mounting the writeable Snapshot for drive X as drive Z from the Snapshot named "new."

```
C:\Program Files\SnapDrive> sdcli snap mount -k Z -d new -d X
```

2. Once the new drive letter is available, take the affected tablespace offline or shut the database down.

```
SQL> alter tablespace users offline
Tablespace altered.
```

3. Use the Explorer or another utility to copy the database file from the mounted Snapshot to the live drive. Regardless of the utility chosen to do the copy, make sure that the permissions are preserved on the file.

4. Initiate Oracle recovery to apply the changes to the data file.

```
SQL> recover tablespace users
Media recovery complete.
```

5. Bring the tablespace back online.

```
SQL> alter tablespace users online
Tablespace altered.
```

6. Unmount the Snapshot to prevent problems with the backup.

```
C:\Program Files\SnapDrive> sdcli snap unmount -d Z
```

The use of the SnapDrive command line interface to present the writeable LUN to the system saves time and considerably reduces the number of steps to restore a data file, thereby reducing the chance of error. The Snapshot can be made available to the system much faster than bringing the file from tape and with less administrative effort. These two characteristics combine to provide a very effective backup and restore solution.

5.2. Restoring All Data Files in a Single File System or a Raw Device

Using NetApp controller LUNs, it is possible to restore a single raw device or device containing a file system. For this method to be implemented successfully, it is imperative that the database be run in archivelog mode. Doing this operation requires that the tablespace on the raw devices or all tablespaces with data files in the file system on the device be taken offline. All data files within the file system will revert back to the Snapshot. Note that only the LUN is restored; other LUNs within the same volume will not be affected.

Recovery using this method is a manual process. The database administrator and the system administrator should work as a team to make sure that each step of the process is followed carefully. In

this example, we are going to restore the tablespace "users," which has its data file located on drive X:. The steps are illustrated below:

1. Take the affected tablespaces offline. Affected tablespaces can be found using the data dictionary by querying for the drive letter assigned to the LUN to be restored. Alternatively, the database can be shut down.

```
2. SQL> select tablespace_name, file_name from dba_data_files where
3.     file_name like 'X:%';
```

```
4. TABLESPACE_NAME          FILE_NAME
5. -----
6. USERS                      X:\ORADATA\ORCL\USERS01.DBF
```

```
7. SQL> alter tablespace users offline;
8. Tablespace altered.
```

The database should not remain online if the system tablespace is affected. The database should be shut down first and offline recovery should be performed.

9. Restore the LUN from the controller's command line.

```
10. C:\Program Files\SnapDrive> sdcli snap restore -d X -s new
11. Initializing...
12. KIM : Checking Policies ...
13. KIM : Success in Checking Policies
14. KIM : Dismounting Volume on the disk ...
15. KIM : Success in Dismounting Volume on the disk ...
16. KIM : Restoring disk snapshot ...
17. KIM : Success in Restoring disk snapshot
18. KIM : Re-Mounting Volume on the disk ...
19. KIM : Success in Re-Mounting Volume on the disk
```

20. Recover the tablespace and bring it online.

```
21. SQL> recover tablespace users;
22. Media recovery complete.
23. SQL> alter tablespace users online;
```

Once the tablespace has been brought online, the data will be intact and available. While not applicable to all situations, this method is extremely fast and presents little impact if the business concerns can withstand the few minutes of data unavailability.

5.3. Restoring a Database That Was Run in Noarchivelog Mode

If your database was running in noarchivelog mode at the time of the failure, point-in-time recovery is the only option. You simply recover the database and all associated data files up to the time when the last cold backup was made.

Recovery is a manual process. The database administrator and system administrator should work as a team during each step of this process. For this reason, we do not supply a set of canned scripts to do a recovery. Instead, we illustrate the steps involved.

The steps for point-in-time recovery are as follows:

1. Shut down the database with the shutdown abort command (if the database is still running).
2. SQL> **shutdown abort**

3. Oracle instance shutdown.
4. Restore the data files (including the control file) from the last Snapshot taken using the snap restore command.
5. C:\Program Files\SnapDrive> **sdcli snap restore -s new -d X**
6. Restart the database.
7. SQL> **CONNECT sys/change_on_install as sysdba**
8. SQL> **STARTUP**

The time required to do a recovery is minimal, only two or three minutes. This is true regardless of the size of the database, since the act of restoring a Snapshot is merely the copying of a single 4Kb block. This allows the database administrator to recover a database that can be many gigabytes in size very quickly.

5.4. Restoring a Database That Was Run in Archivelog Mode

Restoring a database that was running in archivelog mode at the time of the failure presents a much greater range of options. Depending on the needs of the moment, the database administrator can do a point-of-failure recovery, a cancel-based recovery, or a point-in-time recovery. You can also restore individual tablespaces or files. If the affected tablespaces or files are taken offline, you can even do the recovery while the database is open. The full range of techniques covering all of these options is far beyond the scope of this technical report. The example shown herein assumes that the database administrator wishes to perform a point-of-failure recovery of all tablespaces while the database is closed. (That is, the recovery will take the database forward to the point where the failure occurred.) The current control file and all archived and online redo log files are assumed to be available. The most current backup of the data files is assumed to be in the Snapshot called "new." As stated earlier, the database must be stored on two volumes in order to support point-of-failure recovery using the snap restore command. One volume should contain the data files, and the other volume should contain the control file and the log files (including the archived log files). This allows you to run the snap restore command against the data file volume only and thus preserve your control and log files intact.

Recovery is a manual process. The database administrator should be intimately involved in each step of this process. For this reason, we do not supply a set of canned scripts to do a recovery. Instead, we illustrate the steps involved.

The steps for point-of-failure recovery are as follows:

1. Shut down the database with the shutdown abort command (if the database is still running).
2. SQL> **CONNECT sys/change_on_install as sysdba**
3. Connected.
SQL> **SHUTDOWN ABORT**
4. Restore the data files from the last Snapshot taken using the snap restore command. A current copy of the control file should be stored on the volume where the log files are located. The volume containing the archive logs, control files, and redo logs should not be restored or data will be lost.
C:\Program Files\SnapDrive> **sdcli snap restore -d X -s new**
5. Mount the database using the current control file referred to above.

6. SQL> **CONNECT sys/change_on_install as sysdba**
7. Connected.
SQL> **STARTUP MOUNT**
8. Recover the database from the archived and online redo log files.
SQL> **RECOVER AUTOMATIC DATABASE;**
9. Open the database for user access.
SQL> **ALTER DATABASE OPEN;**

At this point, the database should be running in normal mode, with all data intact up to the point of failure.

6. Conclusions

A NetApp controller offers the Oracle database administrator compelling advantages in terms of backup and recovery. Use of Snapshots, combined with conventional backup-to-tape techniques, can dramatically optimize the Oracle database backup operation. Retaining a number of online Snapshots allows the database administrator to restore the database without the necessity to restore from tape in many circumstances. Backup and recovery performance is dramatically improved over conventional local disk and SAN configurations.

7. Caveats

For more information regarding the use of NetApp controllers with Oracle, see Oracle9i for Windows 2000: Integrating with a NetApp Controller in a SAN Environment by Toby Creek. You can also find support on [Oracle's Web site](#) for NetApp controllers. However, NetApp has not tested this procedure with all of the combinations of hardware and software options available on Microsoft Windows. There may be significant differences in your configuration that will alter the procedures necessary to accomplish the objectives outlined in this paper. If you find that any of these procedures do not work in your environment, please contact the [author](#) immediately.

ARCHIVED COPY
Contents may be out-of-date